

INVESTIGACIÓN, ENSEÑANZA Y APRENDIZAJE DE PROGRAMACIÓN PARA AMBIENTES MULTIPLATAFORMA, DISTRIBUIDOS, COLABORATIVOS, INTERACTIVOS Y GRÁFICOS

J J Carreón Granados¹

RESUMEN

Se presentan resultados de investigación relacionados con procesos de enseñanza-aprendizaje de diseño de aplicaciones orientadas a móviles, por ejemplo, en la plataforma Android. Esta investigación se apoya en la hipótesis de que la enseñanza-aprendizaje del diseño de estas aplicaciones, es la enseñanza-aprendizaje de la programación de aplicaciones con base en diseño, *batch* (convencionales) e interactivas (principalmente, las del mundo de los móviles), combinando motivación y principios en la programación de juegos, *mashups* (aplicaciones híbridas) y móviles, en un enfoque escalable de la computación.

Lo cual requeriría un currículo general, que con base en la motivación de los estudiantes abarque desde la primaria, hasta la especializada de posgrado, pasando por la *in-house* corporativa; además de textos, ambientes, metodologías y lenguajes de programación innovadores. Finalmente, pero no menos importante, una adecuada capacitación del profesorado.

Es factible, por ejemplo, construir aplicaciones desarrolladas inicialmente para Android, y que funcionen mejor, eventualmente, en sistemas operativos como el iOS de Apple. Estos resultados del proyecto PAPIIT IN102210, Facultad de Ingeniería, UNAM, “Diseño de aplicaciones, distribuidas, interactivas y gráficas, en Android”, (FI UNAM, 2013) si bien limitados, tienen vastos alcances.

ANTECEDENTES

Entre los antecedentes de este proyecto estaba el de que las redes sociales habían supuesto una revolución reciente en lo que la utilización y concepción de Internet; lo cual se refería, no sólo a que habían promovido el surgimiento de muchas y nuevas formas de empresas, sino debido a que habían modificado la manera de comunicarse entre los individuos.

Pues si bien hasta hacía poco, la manera habitual de comunicarse mediante Internet eran las aplicaciones de mensajería instantánea y el correo electrónico, lo que limitaba frecuentemente los contactos a personas cercanas o conocidas, además de que la forma de comunicarse era principalmente mediante texto.

Con la entrada de las redes sociales, el número de contactos por usuario había crecido exponencialmente, motivando el uso de grupos por preferencias afines y la comunicación había pasado a ser más visual y variada con la inclusión de fotos, videos aplicaciones y *widgets* (dispositivos). Por otro lado, las aplicaciones móviles también habían sufrido algunos cambios. Básicamente, los móviles se usaban para llamadas telefónicas, *SMS* (mensajes de texto) y juegos. Gracias a la evolución del hardware, el rango de empleo se había ampliado a varios campos, como el audiovisual o la navegación por Internet.

La mezcla de estos usos había generado una demanda de aplicaciones que permitían realizar estas acciones de manera sencilla e intuitiva haciendo que los móviles saciaran muchas necesidades que habitualmente se satisfacían con las computadoras habituales. De ahí, que no fuera de extrañar que las redes sociales aparecieran en los móviles, dado que dotaban a éstas de una visión más realista de su empleo que el mantener unidos a

¹ [Profesor Titular TC Sistemas Inteligentes. Facultad de Ingeniería, UNAM. \[juan.carreon@gmail.com\]\(mailto:juan.carreon@gmail.com\)](mailto:juan.carreon@gmail.com)

Con formato: Español (España, internacional)

Con formato: Español (España, internacional)

grupos estuvieran donde estuvieran. En esas redes se podía realizar lo mismo que en las redes fijas como intercambiar videos, archivos... y a la vez dotarlas de nueva funcionalidades como la geolocalización.

Hasta entonces las computadoras más empleadas eran los dispositivos como los teléfonos celulares. Por cada computadora en el mundo, existían cuatro teléfonos celulares. De ahí que la forma más fácil de incorporar a los individuos a las TIC, Tecnologías de Información y Comunicaciones, vinculándolas así a la Sociedad de la Información y del Conocimiento, era lograr que dispusieran y emplearan lo mejor posible un teléfono celular, mejor aún un *smartphone* (teléfono inteligente) o una *tablet*.

Por otra parte, si se tenían editores de texto y de imágenes, hojas de cálculo, aplicaciones de mensajería y la mayoría de aplicaciones que habiendo funcionado sólo en el escritorio se habían trasladado a la web, ¿cómo es que en ésta no se tenían ambientes de desarrollo de didácticos de calidad?

El proyecto "Diseño de aplicaciones, distribuidas, interactivas y gráficas, en Android", buscó generar la metodología, el material documental, y el software que permitieran diseñar programas, componentes y sistemas de programación distribuida, interactiva y gráfica, en principio, bajo la plataforma Android, proyecto que abarcaría 2010-2012; mediante establecer inicialmente un prototipo como podría ser una red social entre móviles, smartphones y tablets.

Se seleccionó Android debido a sus alcances tecnológicos y educativos. A la vez que se buscaría el desarrollo de aplicaciones en dicha plataforma se verificaría la eficacia de concepciones como la del Programming Languages Team, PLT, (Racket, 2013) en el diseño de programas, componentes y sistemas distribuidos, interactivos y gráficos; eventualmente, desarrollados en ambientes web desde móviles que operaran también como redes sociales. (Krishnamurthi, 2009a)

De ahí que los objetivos del proyecto mencionado fueran, además de los previamente mencionados; probar y difundir una metodología sólida de diseño de programas; formar recursos humanos que manejaran nuevos paradigmas de programación. Y que de preferencia todo lo anterior fuera posible realizarlo desde un ambiente web móvil, que pudiera ser empleado por estudiantes muy jóvenes. (Bootstrap, 2013)

De cumplirse los objetivos planteados, sería factible aprender, asimilar e innovar metodologías de diseño de programas para el cómputo gráfico, interactivo y distribuido, como en la plataforma Android, lo que tendría un impacto significativo en el proceso de enseñanza-aprendizaje de asignaturas, no sólo de los planes y programas de estudio de ingeniería, propiciando que profesores y alumnos fueran más competitivos en su desempeño profesional, sino abarcar una población estudiantil más general.

Entre los antecedentes del proyecto mencionado estaba el que desde inicios de la década pasada se había iniciado la colaboración de académicos nacionales con los del PLT, conduciendo a que la principal obra del PLT desde hace algunos años estuviera traducida al español: *Cómo diseñar programas*, (Carreón, 2009) la que en su segunda edición (Fellesein, 2013) actualmente está en proceso, incorporando entre sus novedades el manejo de programación reactiva, núcleo de la programación para móviles.

De ahí que, al final del primer año del proyecto, en 2010, se dispusiera de más de diez aplicaciones sencillas para móviles mediante el compilador Moby, Racket for Mobile Phones, (Krishnamurthi, 2009) el cual soportaba navegadores *desktop* y *smartphones*, incluyendo Android; entre los alcances de este compilador estaba el de convertirse, eventualmente, en el mediano plazo en el lenguaje *scripting* más importante para móviles.

Debido, entre otras razones, a que las demandas de Moby en cuanto hardware eran más accesibles, además de apoyar plenamente un enfoque educativo completo, que incorporaba aspectos esenciales del cómputo reactivo, central en el desarrollo de *apps*, pequeñas aplicaciones, para móviles. Las *apps* desarrolladas con este compilador abarcaban desde juegos, mapas y diferentes manejadores de eventos: como cambios inerciales, variaciones en el tiempo, localización, aceleración, inclinación, conectividad y mensajería, entre otros.

Se consideró también el análisis y el diseño de aplicaciones, *apps*, bajo la plataforma Android, empleando el lenguaje de programación Java, mediante el ambiente de desarrollo Eclipse y el Android SDK, los cuales entonces representaban la forma más general de desarrollar *apps*. Además de explorar opciones adicionales como la impulsada por Motorola, también mediante Java, Eclipse y el MOTODEV Studio; y una más de Google, mediante su ambiente App Inventor, además de herramientas de desarrollo *cross-plataform* para móviles, como Appcelerator.

Para ello se incorporó, adicionalmente a este aspecto del proyecto, un enfoque de diseño de programas con base en clases y abstracciones que soporten el diseño de software reusable y bibliotecas, considerando la relación entre algoritmos y estructuras de datos, así como técnicas básicas para analizar la complejidad algorítmica. (Proulx, 2011) Con el objetivo de involucrar a estudiantes de diferentes ingenierías, incluyendo a los de ingeniería en computación.

A finales del segundo año de los tres que contempló el proyecto, 2011, se había avanzado significativamente hacia el logro de las contribuciones esperadas del mismo; principalmente, en cómo aprender, enseñar, trabajar e investigar la interactividad, mediante el desarrollo de video-juegos, tanto en cursos curriculares y extracurriculares, incrementándose la colaboración con instituciones, grupos de investigación y académicos internacionales y nacionales, así como con una red en el ámbito del cómputo móvil de desarrolladores y empresas internacionales y nacionales. (Carreón, 2013)

Fue rebasado el objetivo de establecer un prototipo como podría ser una red social entre móviles, -seleccionando Android debido a que podría tener grandes alcances tecnológicos y educativos, además de que el desarrollo de aplicaciones en dicha plataforma permitiría verificar la eficacia de concepciones como la del PLT, en el diseño de programas, componentes y sistemas distribuidos, interactivos y gráficos; ya que el proyecto del compilador Whalesong, del propio PLT, mejoraba sustancialmente el de Moby, ya que permitía generar *apps web* para una parte del lenguaje Racket, las cuales pueden liberarse tanto en Android como en iOS, al seguir las APIs webapp HTML5. (Yoo, 2013)

Más importante aún era que WeScheme la aplicación más importante desarrollada inicialmente con el compilador Moby y ahora mejorada significativamente por el Whalesong, ampliaba los alcances del currículo *Bootstrap* (Bootstrap, 2013) para estudiantes entre 12 y 16 años de edad, a quienes enseña a programar sus propios videojuegos empleando exclusivamente conceptos algebraicos y geométricos.

La misión de Bootstrap se apoya en la motivación y confianza de los jóvenes en relación a los videojuegos a fin de **aplicarlos directamente al álgebra para crear algo muy atractivo** para el aprendizaje de las matemáticas y la programación. (Ghose, 2012) Al trabajar con escuelas, distritos escolares y programas educativos a lo largo y ancho de Estados Unidos, primero; y luego de otros países, como India e Indonesia.

A diferencia de la mayoría de las clases de programación, Bootstrap emplea álgebra como medio para crear imágenes y animaciones, lo que significa que los conceptos que los estudiantes encuentran en Bootstrap se comportan de la misma forma que los que encuentran en las clases de matemáticas. Permitiendo a los estudiantes experimentar con conceptos algebraicos mediante escribir programas que permiten a la animación de un avión volar, mediante ecuaciones lineales; que responda a teclazos, o modificarlo cuando choque con otro objeto, empleando fórmulas como la de distancia entre dos puntos. Y, en general, usando cualquier concepto matemático como tarea de programación.

Los estudiantes no sólo aprenden a escribir código, lo que no los convierte automáticamente en programadores o ingenieros en software por el sólo hecho de aprender un lenguaje de programación, sino que aprenden sólidas habilidades de programación, tales como diseñar tipos de entrada y salida, escribir casos de prueba, y explicar su código a otros. Al construir un currículo con esos elementos permite a los estudiantes transitar suavemente de la especificación verbal de un problema al código final que lo resuelve.

Es así que esas habilidades pueden emplearse en otras actividades, incluyendo las de operar con otros ambientes y lenguajes de programación diferentes o más avanzados, y/o cualquier actividad que requiera del análisis de problemas, expresar su esencia de forma abstracta y con ejemplos, formular afirmaciones y comentarios en un lenguaje preciso, evaluar y revisar estas actividades a la luz de verificaciones y pruebas, así como atender cuidadosamente a los detalles. (Fellesein, 2003)

Las concepciones, metodologías y herramientas del PLT están plasmadas en libros, lenguajes de programación, listas de correo, canales de discusión, conferencias anuales, talleres, blogs, cuentas en Twitter, páginas wiki, videos y repositorios de desarrollo en GitHub, entre otros. (Racket, 2013)

Un aspecto eje de las concepciones del PLT es la Programación mediante Diseño, Program by Design, (Program, 2013) un proyecto orientado a la educación en cómputo que emplea contextos atractivos, p e, un programa "Hola Mundo" como una animación, en el cual los estudiantes tienen la oportunidad de programar juegos, "mashups", móviles, entre otros, a la vez que aprenden un enfoque escalable de la programación con base en sólidos principios. A través de un currículo que abarca desde la primaria, hasta el posgrado, pasando por la capacitación corporativa "in-house", conduciendo a innovaciones en ambientes, metodología y lenguajes de programación.

ANÁLISIS

Si se hace un *zoom-back* hacia el sistema educativo nacional, es fácil verificar que prácticamente se ha duplicado el número de egresados de la educación media superior en México en los últimos años, sin embargo, el sistema educativo aún enfrenta retos enormes en lo referente a la matrícula de los estudiantes, su rendimiento y la transición al mercado laboral. (OCDE, 2012)

Según el Programa Internacional para la Evaluación de Estudiantes, PISA, el rendimiento en matemáticas de los alumnos mexicanos si bien mostró cierta mejora entre 2006 y 2009, sigue siendo bajo en comparación con otros países de la OCDE. Lo cual es importante no sólo a fin de asegurar que en la educación superior ingresen estudiantes con buenas probabilidades de éxito, sino también desde la perspectiva de mantener y mejorar la tendencia de los estudiantes a estudiar ingeniería. (OCDE, 2012)

Lo que ha originado actividades crecientes orientadas a mejorar la calidad en la educación básica y media, abarcando desde la investigación acerca de este tema, hasta la formación de maestros de primaria y secundaria orientada a brindar a las nuevas generaciones ambientes de aprendizaje más efectivos y atractivos que formen no sólo a futuros candidatos a estudiar ciencia e ingeniería, sino ciudadanos con los elementos básicos de alfabetización para la época actual. (Duque, 2011)

En un contexto general en el que se requiere tomar medidas concretas para apoyar mejor a las escuelas y a los docentes en el cumplimiento de sus tareas, dar mayor impulso a la educación y formación tecnológicas y hacer que la educación superior sea más accesible y pertinente para un mayor número de estudiantes.

Y en donde lograr ambientes de aprendizaje eficaces, implica emplear estrategias de enseñanza adecuadas a los objetivos diseñados, en el marco de un currículo coherente con base en materiales educativos apropiados, en una organización capaz de apasionarse e incentivar la docencia de calidad, con estudiantes con los conocimientos y las habilidades requeridas para aprender exitosamente. Además de requerirse de forma creciente mejorar los procesos de formación de ingenieros empleando ciencia y tecnología.

En relación a esto último, existen esfuerzos institucionales para incidir tanto en la superación de deficiencias en los antecedentes con los que ingresan los estudiantes a las facultades y escuelas de ingeniería, como para lograr ambientes de aprendizaje más eficaces, mediante foros acerca de la enseñanza de la física, las matemáticas, las ciencias básicas, la informática y los objetos virtuales, el diseño y la construcción de modelos y prototipos experimentales, así como mediante concursos. (Memoria, 2013)

Además de otros medios, como asesorías presenciales y en línea, talleres de ejercicios, conferencias y conferencias-clase, seminarios de robótica aplicada al aprendizaje de la ciencia y la tecnología, centros de recursos de aprendizaje para las ciencias básicas (recursos web de material didáctico digital), así como proyectos de investigación que buscan incorporar Tecnologías de Información y Comunicación TIC's en el proceso enseñanza aprendizaje de las asignaturas de ciencias básicas. (CB, 2013) Además de incluir en propuestas de actualización de planes y programas de estudio que el aprendizaje de la programación debe ser desde los primeros semestres.

Sin embargo, existe la percepción de que para la enseñanza, el aprendizaje e investigación aún no se enfatizan suficientemente las nuevas tecnologías de información y comunicación, TIC, a pesar de que en las principales universidades del mundo para algunas áreas de ingeniería incluso existen tendencias como las de desaparecer laboratorios de enseñanza y sustituirlos por apoyos con base en TIC, constituyéndose como parte de los recursos que propician el aprendizaje autónomo de los estudiantes, (Zabalza, 2013) que los capacite para el aprendizaje de por vida. (CT, 2012)

Dos aspectos surgen en relación a los esfuerzos mencionados: si bien los comités de carrera de algunas instituciones (CT, 2012) comienzan a hacer converger sus requerimientos con los de ciencias básicas, esa convergencia aún no es suficientemente satisfactoria como para que exista como identidad de objetivos; principalmente, entre el área que proporciona los servicios de formación en cómputo a estudiantes de las diversas ingenierías, y las áreas de formación en ciencias básicas y en ciencias de la ingeniería.

Independientemente de que sea complejo explicar las razones de esa disonancia... parece que no es un problema exclusivo de algunas instituciones, inclusive se pueden dar buenos argumentos de que se trata de un fenómeno global, o que hasta hace poco se consideraba así, el cual al final de cuentas se expresaba en juicios como los de que "los mejores estudiantes de las mejores universidades a nivel mundial cuando egresan no programan, incluyendo a los de computación." O como afirma un ejecutivo de SAP.

Wir sind froh, dass die Absolventen schon Java können. Programmieren müssen wir denen halt noch beibringen. *(Sperber, 2013)*

Si bien este es un problema común nacional e internacional relacionado con la elevación de la calidad de la educación superior en ingeniería que implicaría cuando menos temas como los de redes académicas, desarrollo curricular, mejoramiento del profesorado, educación a distancia y empleo de tecnologías de la información y comunicaciones.

Ese problema requiere ubicarse primero en el contexto de la educación básica, primaria y secundaria, en la que urge eleve su calidad y supere sus rezagos; una educación media y superior con una cobertura de sólo del 60%, que requiere integrar muchos de sus contenidos con métodos didácticos intensivos en TIC y en capacidades laborales; y una educación superior con una cobertura aproximada aún menor del 32%, donde la educación media superior tiene una eficiencia terminal del 55% y la superior de 50%. Con la consiguiente elevación de costos para los educandos, sus familias, el país, lo cual afecta la calidad de vida de los no graduados y de las generaciones futuras, afirma Reséndiz (2013), citando (OCDE, 2012)

Lo cual significa grandes costos para los individuos y para el país, disminuyendo las opciones de calidad de vida tanto individuales como colectivas. En el caso de las escuelas y facultades de ingeniería, reducir esas afectaciones, significa una mejor formación en las ciencias básicas y las ciencias de la ingeniería.

CONCLUSIONES

Es importante considerar, algunos enfoques en apoyo de los procesos de enseñanza-aprendizaje relacionados con base en la fase actual de la computación: móvil, interactiva, gráfica, distribuida, y en nube, en particular como el de Bootstrap.

Esos procesos de enseñanza-aprendizaje deben y pueden apoyarse en la motivación de los estudiantes, y en sólidos principios computacionales que den soporte a los de ciencias básicas.

Esos procesos deben y pueden abarcar desde la educación básica, hasta los niveles más avanzados de la educación superior.

Esos procesos deben y pueden abarcar diversas plataformas actuales y futuras, tanto desde el aspecto del desarrollo, como de su distribución y aprovechamiento; al abarcar desde aspectos lúdicos, hasta los de investigación y desarrollo.

Esos procesos pueden y deben aprovechar experiencias valiosas de otras latitudes.

Esos procesos pueden y deben contribuir significativamente a mejorar el rendimiento del aprendizaje en general, iniciando con el matemático, y continuando con los propios de la ingeniería y de la ciencia.

Esos procesos pueden y deben contribuir a incrementar la eficacia y eficiencia de los numerosos y valiosos esfuerzos institucionales orientados a superar las deficiencias en antecedentes con los que ingresan los estudiantes a las facultades y escuelas de ingeniería.

Esos procesos pueden y deben darle un nuevo sentido a las propuestas de actualización de planes y programas de estudio que enfatizan que el aprendizaje de la programación debe ser desde los primeros semestres.

Esos procesos pueden y deben apoyar el aprendizaje de la programación con base en el desarrollo de capacidades de interrelación personales.

Esos procesos pueden y deben vincular el aprendizaje de la programación con la solución de problemas en otros dominios tecnológicos y científicos, como el matemático, (apoyándolos, pero no apoyándose en ellos).

Esos procesos pueden y deben apoyarse en la colaboración intra e inter institucional, nacional e internacional.

BIBLIOGRAFÍA

Bootstrap. (2013). *Overview*, Brown University, Providence, RI, USA, <http://www.bootstrapworld.org/>

Bootstrap. (2013). *Teach*, Brown University, Providence, RI, USA, <http://www.bootstrapworld.org/>

Carreón J J. (2009), *Cómo diseñar programas*, FI, UNAM

Carreón J J. (2013). La historia cotidiana del proyecto a lo largo de los últimos cuatro años puede seguirse en twitter: @androidjj, @androidMexico, @ayedd

CB-FI-UNAM. (2013). *Memoria de eventos*, División de Ciencias Básicas, Facultad de Ingeniería, UNAM. <http://dcb.fi-c.unam.mx/Eventos/>

CB-FI-UNAM. (2013). *División de Ciencias Básicas*, FI, UNAM. <http://dcb.fi-c.unam.mx/>

CT. (2012). *Informe de la Comisión de Planes y Programas de Estudio del Consejo Técnico*. Plan de desarrollo, FI, UNAM, 2011-2014/1.1, 21 de noviembre de 2012
<http://www.ingenieria.unam.mx/planeacion/paginas/planes.html>

Duque M, Celis J y Camacho A. (2011). *Cómo lograr alta calidad en la educación de los ingenieros: una visión sistémica*, Revista Educación en Ingeniería, Diciembre de 2011, No 12, pp 48-60, Colombia

Fellesein M. (2003), *Preface, How to design programs*. MIT Press, 2003
<http://htdp.org/2003-09-26/Book/curriculum-Z-H-2.html>

Fellesein M. (2013) *How to Design Programs* (draft), Second Edition, MIT Press, March 8th, 2013, <http://www.ccs.neu.edu/home/matthias/HtDP2e/Draft/index.html>

FI UNAM. (2012). *Plan de desarrollo 2011 - 2014*, FI, UNAM, México

FI UNAM. (2013). *Proyectos PAPIIT vigentes*, Secretaría General, FI, UNAM,
http://sgeneral.fi-a.unam.mx/dies/papime/papiit_vigentes.php

Ghose T. (2012). *Like Math? Thank Your Motivation, Not IQ*. Scientific American, December 28, 2012, http://www.scientificamerican.com/article.cfm?id=like-math-thank-your-moti&WT.mc_id=SA_sharetool_Twitter

Krishnamurthi S. (2009a). *On WeScheme*, vidiowiki, feature, International Lisp Conference, MIT, Cambridge, Mass, <http://www.vidiowiki.com/watch/cydr9yk/#>

Krishnamurthi S. (2009b). *The Moby Scheme Compiler for Smartphones, or, Is That a Parenthesis in Your Pocket?*, International Lisp Conference, MIT, Cambridge, Mass, <http://planet.racket-lang.org/package-source/dyoo/moby.plt/3/10/planet-docs/manual/index.html>

Program by Design. (2013). *Overview*, <http://www.programbydesign.org/>

Proulx V. (2012) *How to Design Classes* (draft), Northeastern University, Boston Mass, <http://www.ccs.neu.edu/home/matthias/HtDC/htdc.pdf>

Racket. (2013) *Community*, PLT se refiere al grupo que está en el núcleo de desarrollo de Racket y el cual consiste de numerosos individuos distribuidos en diferentes universidades, principalmente en Estados Unidos. Racket Community, <http://racket-lang.org/community.html>

Racket. (2013). *Racket Community*, <http://racket-lang.org/community.html>

Reséndiz D. (2013). *El valor y los costos de la educación*, Conferencia, FI, UNAM

Sperber M. (2013) *Overheard in a German firm*, citado en <http://www.ccs.neu.edu/home/matthias/>

Toledo D y Ramos G I. (2012). *Los retos para el sistema educativo*, Getting It Right, Una agenda estratégica para las reformas en México, OCDE Publishing, p 125
<http://www.oecd.org/centrodemexico/Getting%20It%20Right%20EBOOK.pdf>

Yoo D. (2013). *Whalesong: a Racket to JavaScript compiler*,
<http://hashcollision.org/whalesong>

Zabalza M A, *Aprendizaje Autónomo*, Diplomado de docencia en educación superior, Universidad Central, https://www.youtube.com/watch?v=MoOt8H-i_W0